

# 时间敏感网络交换处理逻辑 TSS 设计方案 (版本 1.0)

OpenTSN

OpenTSN 开源项目组

2021 年 5 月

## 版本历史

版本	修订时间	修订内容	修订人	文件标识
1.0	2021.5	初版编制		OpenTSN3.0

## 目录

1、概述 .....	4
2、总体设计 .....	4
2.1. 总体架构 .....	4
2.1.1 网络输入处理逻辑 .....	5
2.1.2 网络输出处理逻辑 .....	7
2.1.3 内部处理逻辑 .....	9
2.2 帧的处理流程 .....	11
附录 A：数据格式定义 .....	13
附录 B：内部表项格式定义 .....	16
附录 C：内部寄存器定义 .....	17
附录 D：NMAC 报文格式 .....	23
附录 E：command/command_ack 命令格式 .....	26

## 1、概述

OpenTSN2.0 开源硬件逻辑既可作为 TSN 网卡又可作为 TSN 交换机使用，为了简化其作为 TSN 交换机使用时的逻辑复杂度以及增强其作为 TSN 网卡使用时的功能可扩展性，并且考虑到逻辑模块的复用，现将 OpenTSN2.0 开源硬件逻辑拆分为时间敏感网络硬件控制逻辑（HCP）、时间敏感网络端处理逻辑(TSE)、时间敏感网络交换处理逻辑(TSS)三个模块。本文详细介绍时间敏感网络交换处理逻辑 TSS 的设计方案，TSS 主要负责对报文进行交换并基于 IEEE 802.1Qbv/IEEE 802.1Qch 对时间敏感流量进行调度。

## 2、总体设计

### 2.1. 总体架构

时间敏感网络交换处理逻辑的总体架构框图如图 2-1，其中 GMII0~GMII3 接口为网络交换接口，GMII4 接口为连接 HCP 模块的专用接口。

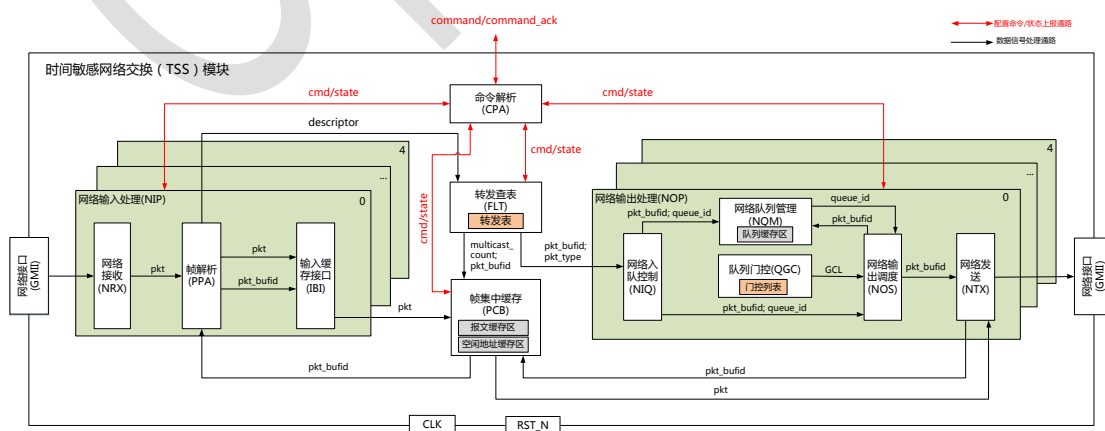


图 2-1 总体架构框图

总体架构图中的信号格式含义如下表 2-1。

表 2-1 TSS 总体架构信号定义

信号	位宽	含义
pkt_134（报文集中缓存接收或发送的信号）	134	报文体数据，具体格式参考附录 A
pkt_9（非报文集中缓存模块的信号）	9	报文体数据，具体格式参考附录 A
descriptor	46	报文描述符数据，用于标识报文数据
pkt_bufid	9	报文数据在报文缓存区中缓存的 ID 号
pkt_type	3	报文类型数据
flow_id	14	报文数据的流 ID，用于标识流
lookup_en	1	查表使能
outport	9	输出端口号（bitmap）
inport	4	输入端口 号
queue_id	3	queue_id 号，用于标识队列缓存的 ID 号
multicast_count	4	组播报文的输出端口数量
GCL(gate_ctrl_vector)	8	门控信号
time_offset	49	1588 同步从时钟架构需要补偿的值
command	204	读、写命令
command_ack	204	读命令响应

现将整个架构划分为三大部分逻辑：网络输入处理逻辑、网络输出处理逻辑以及内部处理逻辑，下面对此三部分逻辑进行逐一介绍。

### 2.1.1 网络输入处理逻辑

网络输入处理逻辑的内部组成框图如图 2-2。

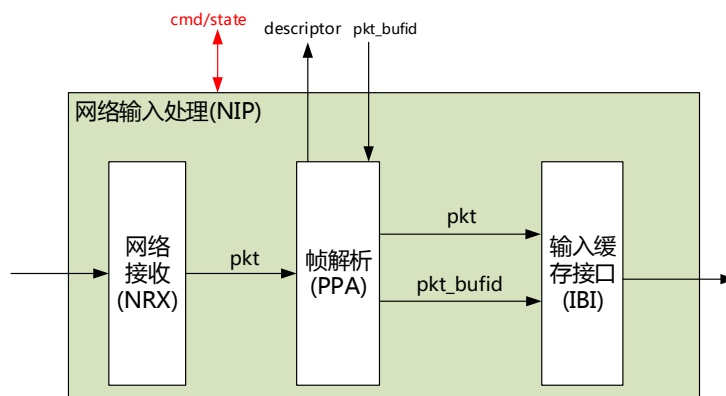


图 2-2 NIP 模块组成框图

**NRX(Network RX, 网络接收)模块：**主要功能是接收网络接口传来的报文，将报文传输时钟域从 GMII 接口时钟域切换到架构内部时钟域，以及记录架构接收时间同步报文的时间，并存放在 TSNTag 中。根据交换处理逻辑所处的阶段来决定是否传输报文：若交换处理逻辑处于初始化阶段，丢弃所有报文；若交换处理逻辑处于配置阶段，只传输配置报文；若交换处理逻辑处于时钟同步阶段，传输非时间敏感报文；若交换处理逻辑处于正常工作阶段，接收所有报文。

**FPA(Frame Parse, 帧解析)模块：**主要功能是将报文每拍数据位宽由 9bit 转换为 134bit，构造报文描述符，对报文进行监管。在构造报文描述符时，需要根据接口类型来判断该报文是否经过映射，若接口为非合作类型，则该报文未经映射，构造报文描述符使其从主机接口输出，若接口为合作类型，则该报文已经映射，从 TSNTag 中提取相关信息构造报文描述符。报文监管是根据 bufid 剩余数量与报文阈值的大小关系来实现的：若 bufid 剩余数量少于 RC 报文阈值时，将 RC 报文和 BE 报文均丢弃，若 bufid 剩余数量少于 BE 报文阈值时，将 BE 报文丢弃，若 bufid 剩余数量少于未经映射报文阈值时，将未

经映射报文丢弃。

**IBI(Input Buffer Interface, 输入缓存接口)模块：**主要功能是将报文数据发送给报文集中缓存模块进行缓存。本模块接收帧解析模块传来的 134bit 数据，并使用两个寄存器进行缓存，其中任何一个寄存器有数据则往报文集中缓存模块发出写请求，在接收到报文集中缓存模块传来的响应后，才完成一拍数据的写入；并将帧解析模块传来的数据写入另一个寄存器中。

### 2.1.2 网络输出处理逻辑

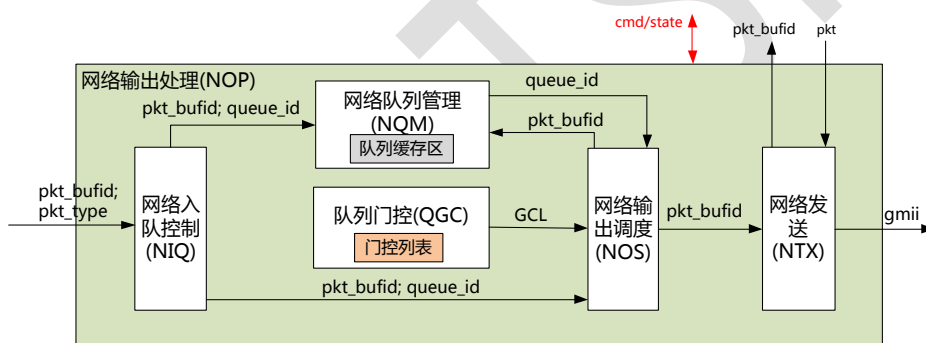


图 2-3 NOP 模块组成框图

**NIQ(Network Input Queue, 网络入队控制)模块：**主要功能是将pkt\_bufid 写入到网络队列管理模块中进行缓存。本模块需要根据接收到的报文类型信息、队列门控模块传来的门控信息进行 queue\_id 的映射，并将pkt\_bufid 与 queue\_id 发送给网络队列管理模块进行缓存。同时将 pkt\_bufid 与 queue\_id 发送给网络输出调度模块，以便对队列首地址进行管理。本模块还需要根据写入队列的信息与网络输出调度模块传输的调度队列的信号来对队列的状态进行管理，主要是对所有队列中目前所写入的 bufid 数量进行管理。

**NQM (Network Queue Manage, 网络队列管理) 模块:** 主要功能是对网络输出端口的队列集中缓存管理

**QGC(Queue Gate Control, 队列门控)模块:** 主要功能是根据架构的全局时钟进行门控列表的读取, 并将门控列表(门控列表的格式如下表 2-2)中 8 个队列的门控开关信息发送给网络入队控制模块和网络输出调度模块。

表 2-2 门控列表数据格式

名称	含义	备注
gate_ctrl_vector[7:0]	门控向量, 对应 8 个门控信息。	使用 RAM 进行实现, 深度为 1024

**NOS(Network Output Schedule, 网络输出调度)模块:** 主要功能是根据调度后的队列信息从网络队列管理模块提取出 pkt\_bufid。本模块需要根据当前队列信息与队列门控模块传来的门控信息进行计算, 得出一个最优先调度的队列, 并从网络队列管理模块的对应队列中得到 pkt\_bufid。

**NTX(Network TX, 网络发送)模块:** 主要功能是从报文缓存区中读取报文并释放 pkt\_bufid、计算时间同步报文的透明时钟并累加到透明时钟域、将报文传输时钟域从架构内部处理时钟域切换到 GMII 发送时钟域、构造帧前导符和帧开始符并将报文从网络接口输出。读取报文时, 需要先将 pkt\_bufid 映射成报文在集中缓存区中缓存的基地址, 并根据此地址往报文集中缓存模块读取报文, 将读出报文的位宽由 134bit 转换为 8bit, 在接收到最后一拍数据时, 将此 pkt\_bufid 传输给报文集中缓存模块进行释放, 以便后续进入架构的报文使用。



### 2.1.3 内部处理逻辑

**FLT(Forward Lookup Table, 转发查表)模块：**主要功能是根据接收到的报文描述符提取出 flow\_id, 根据 flow\_id 进行查找转发表(转发表的数据格式如下表 2-3), 根据查表得到的输出端口进行 pkt\_bufid 和 pkt\_type 的转发。当描述符中查表使能信号 (lookup\_en) 为低 (代表本描述符不需要查表) 时, 直接将描述符中的输出端口 (outport) 字段作为输出端口号进行转发。当得到的输出端口多位为高时, 意味处理的描述符对应报文是个多播报文, 需要将 pkt\_bufid 和 pkt\_type 往多个端口进行转发, 同时将 pkt\_bufid 与输出端口的数量发送给报文集中缓存模块进行缓存以便进行组播计数。

表 2-3 转发表数据格式

名称	含义	备注
outport[8:0]	输出端口号(bitmap), 总共 8 个网络接口与 1 个主机接口。	使用 RAM 进行实现, 深度为 16K

**PCB(Pkt Centralize Buffer, 报文集中缓存)模块：**主要功能是对架构需要转发的所有报文进行集中缓存 (报文缓存区的格式如下表 2-4), 对空闲地址缓存区 pkt\_bufid 进行缓存管理 (空闲地址缓存区的格式如下表 2-5)。每个 pkt\_bufid 都需要设计一个计数器来对输出端口数量进行计数, pkt\_bufid 每释放一次, 将计数器减一; 当 pkt\_bufid 使用完进行释放的时候需要检测该 pkt\_bufid 对应计数器的值, 只有当计数器值为 0 才能进行将 pkt\_bufid 写入空闲地址缓存区中, 当不为 0 意味着该 pkt\_bufid 存着一个组播报文并且该报文还未从所有需要输出的端口输出。

报文缓存区将 1024KB 的空间划分成 512 个报文缓存块，每个报文缓存块能缓存一个 2KB 的报文。

表 2-4 报文缓存区数据格式

地址[8:0]	内容[133:0]
0-127	第 1 个报文缓存块
128-255	第 2 个报文缓存块
...	..
65408-65535	第 512 个报文缓存块

表 2-5 空闲地址缓存区数据格式

名称	含义	备注
pkt_bufid[8:0]	当前“报文缓存区”中空闲的报文缓存块 ID 号。	使用 RAM 进行实现，深度为 512。

**CPA(Command Parse, 命令解析)模块：**负责将接收到 HCP 的 command 命令（命令数据格式见表 2-6）进行解析，来实现对本地寄存器、转发表、门控列表的配置；根据接收到的读命令，将读数据封装在响应中输出给 HCP 模块。

表 2-6 command/command\_ack 命令格式

位置	位宽	名称	说明
[203:180]	8	node_id	该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。
[179:172]	8	dest_module_id	该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID
[171:168]	4	type	4'b0001:寄存器或表项的写命令； 4'b0010:寄存器或表项的读命令； 4'b0110:寄存器或表项的读响应。
[167:152]	16	addr	寄存器或表项的读/写地址
[151:0]	152	data	寄存器或表项的读/写数据；其中五元组映射表的表项位宽最大，为 152bit

## 2.2 帧的处理流程

每个网口配置的端口类型皆为合作类型，所有从网口进入的报文皆为映射后的报文，且都是通过查表去确认输出端口，报文的详细处理流程如下：

- 1) 报文由 p0-p3 网口 (GMII\_RX) 进入网络接收模块，完成 GMII 时钟域到架构内部处理时钟域的转换，并记录报文接收时间戳；然后根据交换处理逻辑所处的阶段来决定是否传输报文：若交换处理逻辑处于初始化阶段，丢弃所有报文；若交换处理逻辑处于配置阶段，只传输配置报文；若交换处理逻辑处于时钟同步阶段，传输非时间敏感报文；若交换处理逻辑处于正常工作阶段，传输所有报文。将报文往帧解析模块发送；
- 2) 帧解析模块收到报文后，对报文进行解析，提取报文的特征信息 (pkt\_type、Flow id、submit addr、Lookup\_en、Outport) 与报文集中缓存模块分配的 pkt\_bufid 并构造成描述符，然后根据 bufid 剩余数量与报文阈值的大小关系来对报文进行监管：若 bufid 剩余数量少于 RC 报文阈值时，将 RC 报文和 BE 报文均丢弃，若 bufid 剩余数量少于 BE 报文阈值时，将 BE 报文丢弃。将报文描述符发送到转发查表模块；同时帧解析模块将 8bit 的数据写入到 134bit 的寄存器中，当一个寄存器写满了 134bit 便将寄存器中的数据传输到输入缓存接口模块，完成报文的位宽转换；

- 3) 输入缓存接口模块接收 134bit 的分组数据, 因报文集中缓存区是采用分时复用的方式进行数据的写入, 因此输入缓存接口模块需要等到报文集中缓存模块返回一个确认信号之后才能进行下一次的数据写入。当其中一个寄存器写满, 且还未收到报文集中缓存模块返回的确认信号时, 后续传来的数据就写入到另一个寄存器, 每个寄存器写满再发送给报文集中缓存模块进行报文的缓存, 以此轮询, 直到报文全部写入完成;
- 4) 转发查表模块会根据报文描述符的 `lookup_en` 信号判断该报文是否需要查表转发, 若不需要查表, 则直接将报文描述符中的 `outport` 信息作为转发的输出端口, 将 `pkt_bufid` 与 `pkt_type` 信息发送给对应输出端口逻辑; 若需要查表, 则从报文描述符中提取出 `flow_id` 进行查找转发表, 得到输出端口号, 并将 `pkt_bufid` 与 `pkt_type` 信息发送给对应输出端口逻辑;
- 5) 输出端口逻辑中的网络入队控制模块接收到 `pkt_bufid` 与 `pkt_type` 后, 根据队列门控模块的门控信息与 `pkt_type` 进行缓存队列的选择, 并将选择好的 `queue_id` 与 `pkt_bufid` 一起发送给网络队列管理模块进行缓存。
- 6) 网络输出调度模块根据队列门控模块的门控信息以及每个队列的状态信息计算出最优先调度的 `queue_id`, 并将此 `queue_id` 中的第一个 `pkt_bufid` 缓存的地址发送给网络队列管理模块, 等待网络队列管理模块将对应的队列中的数据读取出来, 将第一个 `pkt_bufid` 的地址作为 `pkt_bufid` 发送给网络发送模块, 并

根据读取出来的数据更新此 queue\_id 中第一个 pkt\_bufid 所缓存的地址。

- 7) 网络输出调度模块接收到 pkt\_bufid 之后发送给网络发送模块，网络发送模块根据 pkt\_bufid 往报文集中缓存模块进行报文的提取。网络发送模块内部维持两个寄存器，依次将 134bit 的数据转换成 8bit 的数据往后发送，跨时钟域处理后往 p0-p3 接口（GMII\_TX）输出；当一个寄存器读空之后再往报文集中缓存模块进行下一个 134bit 数据的读取，直到报文数据全部读取完成。

## 附录 A：数据格式定义

### ● TSNTag 格式

在流量发送端的网卡内部需要根据报文七元组（目的 mac、type、IP 五元组）对时间敏感、带宽预约、尽力转发流量进行分类映射。将分类映射的结果与原报文的 DMAC 字段进行替换，以此进行 TSN 网络的交换，直到接收端的网卡内部进行 DMAC 还原。被替换的 DMAC 字段被定义成 TSNTag。

表 A-1 分类映射关键字 Key

位宽	名称	描述
48	DMAC	报文目的 MAC
16	ETHTYPE	报文以太网类型
8	protocol	报文协议类型
32	Sip	报文源 ip
32	Dip	报文目的 ip

16	Sport	报文源端口
16	Dport	报文目的端口

因同步报文的 TSNTag 中“seq\_id”、“frag\_id”、“inject\_addr”、“submit\_addr”信息是无用的，因此可以将时间同步报文的这些字段用来存放架构的接收时间戳信息。而其他非时间同步报文的架构接收时间戳信息是无用的，因此可以延用这些字段的信息。

表 A-2 时间同步报文的 TSNTag

位宽	名称	位置	描述
3	Flow type	[47:45]	流类型。100：同步报文（其他报文的格式如表 A-3）
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。
12	Reserve	[30:19]	保留
19	Rx_timestamps	[18:0]	架构接收到时间同步报文的本地时间信息，用于架构发送报文时计算透明时钟。

表 A-3 非时间同步报文的 TSNTag

位宽	名称	位置	描述
3	Flow type	[47:45]	流类型。 000:ST 分组 001:ST 分组 010: ST 分组 011: RC 分组 101: NMAC 分组 110: BE 分组 111: BE 分组
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。
16	Seq id	[30:15]	用于标识每条流中报文的序列号
1	Frag flag	[14]	用于标识分片后的尾。0：分片后的中间报文 1：尾拍
4	Frag ID	[13:10]	用于表示当前分片报文在原报文中的分片序列号
5	inject addr	[9:5]	ST 流在源端等待发送调度时缓存地址

5	submit addr	[4:0]	ST 流在终端等待接收调度时缓存地址
---	-------------	-------	--------------------

### ● 内部传输的 pkt 数据格式

在模块之间传输的有两种格式，一种位宽为 9bit；另一种位宽为 134bit。具体区别如下：

pkt\_data 位宽为 9 位，包含 1bit 头尾标志、8bit 报文数据。头尾标志：1 表示报文头/尾数据；0 标识报文体中间数据。具体如图 A-1 所示。

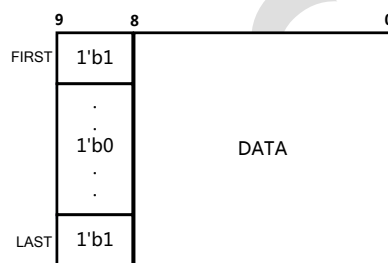


图 A-1 pkt\_data 数据格式 1

pkt\_data 位宽为 134 位，包含 2bit 头尾标志、4bit 无效字节、128bit 报文数据。头尾标志：01 表示报文头；11 表示报文体中间数据；10 表示报文尾。无效字节用于标识报文尾中无效的字节数。具体如图 A-2 所示：

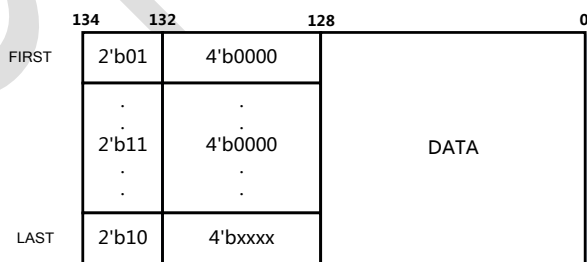


图 A-2 pkt\_data 数据格式 2



## 附录 B：内部表项格式定义

### ● 转发表

转发表的格式如表 B-1 所示。

表 B-1 转发表数据格式

名称	含义	备注
outport[8:0]	输出端口号(bitmap)，总共 8 个网络接口与 1 个主机接口。	使用 RAM 进行实现，深度为 16K

转发表设计在转发查表(FLT)模块，该模块会根据接收到的描述符提取出 flow\_id，再将 flow\_id 作为查表地址进行查找转发表。查表结果为输出端口号，其中某位为高则代表往某个对应的输出端口进行转发，本模块根据查表结果将描述符内容转发给对应的输出端口。转发表的深度为 16K，代表由 TSNSwifth 组成的整个网络最多能够支持 16K 条流的转发。

### ● 门控表

门控表的格式如表 B-2 所示：

表 B-2 门控表数据格式

名称	含义	备注
gate_ctrl_vector[7:0]	门控向量，对应 8 个门控信息。	使用 RAM 进行实现，深度为 1024

门控表设计在交换发送处理逻辑中的队列门控(QGC)模块，该模块根据接收到的时间槽信息以及时间槽切换信号来进行查表。本模块每接收到一个时间槽切换信号，便将时间槽的 ID 作为查表地址进行查找门控表，查表结果为门控向量，对应 8 个门控信号，当某位为高则代表打开对应的某个队列的门控，网络输出调度(NOS)模块根据该



门控进行对队列进行调度。门控表的深度为 1024，代表 TSNSwitch 最多能够支持 1024 个时间槽做为一个周期。

## 附录 C：内部寄存器定义

架构内部可配置地址空间主要有两部分，包括：MDID 模块号和真实地址空间，其中 MDID 模块号主要用来区分不同模块，而后 20 位为各个模块使用的地址空间。地址的第 19bit 位用于区别地址类型，控制/表项寄存器可读可写，调试和版本寄存器只读，每个模块的地址空间为 1024k,其中可读可写和只读寄存器各有 512k。具体地址含义如下。

表 C-1 地址格式

ADDR[26:0]		
MDID[26:20]	ADDR[19]	ADDR[18:0]
MDID : 0-127	0	该模块的控制寄存器,表项等,可读可写
	1	只读

每个处理模块的 MDID 号分配如下：

表 C-2 模块中的 MDID 和地址

处理模块	CSM	TIS	TSS	QGC	GTS	FLT
MDID	0x0	0x1	0x2	0x3-0xa	0xb	0xc
地址	0x0-0xffff	0x100000-0x1fffff	0x200000-0x2fffff	0x300000-0xafffff	0xb00000-0xbfffff	0xc00000-0xcfffff

### ● CSM 模块

地址范围为 Addr 0x0-0xffff。

表 C-3 CSM 模块寄存器

Addr	Data
------	------

	[31:24]	[23:16]	[15:8]	[7:0]
0x0	offset_l			
0x1	offset_h			
0x2	time_slot			
0x3	cfg_finish			
0x4	port_type			
0x5	qbv_or_ach			
0x6	report_type			
0x7	report_en			
0x8	inject_slot_period			
0x9	submit_slot_period			
0xa	report_period			
0xb	offset_period			
0xc	rc_regulation_value			
0xd	be_regulation_value			
0xe	unmap_regulation_value			
0xf ~ 0xfffff	reserve			

表 C-4 寄存器的具体含义

name	bit	R/W	description	default
offset_l	31:17	R/W	代表时间偏移的高位值的低 15 位，表示毫秒	0
	16:0	R/W	时间偏移的低位，表示拍数	0
offset_h	31:17	R/W	保留位	0
	16	R/W	代表时间偏移的正负值，1 代表正值，如果为 0，则代表负值	0
	15:0	R/W	代表时间偏移的高位值的高 16 位，表示毫秒	0
time_slot	31:11	R/W	保留	
	10:0	R/W	时间槽大小	0
cfg_finish	31:2	R/W	保留	0
	1:0	R/W	配置完成寄存器，0 代表架构正在初始化，不接	0

name	bit	R/W	description	default
			收任何报文， 1 代表初始化完成，可以接收 NMAC 配置报文 2 代表配置完成，可以接收除 ST 报文的任何报文 3 代表可以接收任何报文	
port_type	31:8	R/W	保留	0
	7:0	R/W	网络端口类型寄存器，架构共有 8 个网络端口，寄存器的 0-7 位分别代表 0-7 端口的类型，1 代表非合作类型，处理标准以太网类型的报文，0 代表合作类型，处理 TSN 报文	0
qbv_or_ach	31:1	R/W	保留	0
	0	R/W	调度模式选择信号，网络输出逻辑中的调度机制是 QBV 模式还是 QCH 模式 0 代表 QBV 模式；1 代表 QCH 模式	0
report_type	31:16	R/W	保留	0
	15:0	R/W	上报类型，具体参考附录 D	0
report_en	31:1	R/W	保留	0
	0	R/W	上报使能信号，配置与状态管理模块是否进行周期性上报 0 代表不上报；1 代表上报	0
inject_slot_period	31:12	R/W	保留	0
	10:0	R/W	注入时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个	0
submit_slot_period	31:12	R/W	保留	0
	10:0	R/W	提交时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个	0
report_period	31:12	R/W	保留	0
	11:0	R/W	上报周期，配置与状态管理模块上报的周期值 配置的值范围：1 (ms) 或 1000 (ms)	0
offset_period	31:24	R/W	保留	0

name	bit	R/W	description	default
	23:0	R/W	offset 补偿的配置周期	
rc_regulation_value	31:9	R/W	保留	0
	8:0	R/W	RC 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 RC 报文	
be_regulation_value	31:9	R/W	保留	0
	8:0	R/W	BE 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 BE 报文和 RC 报文	
unmap_regulation_value	31:9	R/W	保留	0
	8:0	R/W	非映射流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃非映射报文	
reserve	31:9	R/W	保留	0

### ● TIS 模块

地址范围为 Addr 0x100000-0x1fffff。

表 C-5 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
send_table_N 0x100000-0x1003ff	ST 报文发送时刻表每项内容，N=0、1、...、1023 send_table_0 表示第 0 个发送表			
0x100400-0x1fffff	保留			

表 C-6 寄存器的具体含义

name	bit	R/W	description	default
send_table_0	16	R/W	保留	0
	15	R/W	表项有效位，0 代表无效，1 代表有效	0
	14:5	R/W	ST 流在一个应用周期内的注入时间槽	0
	4:0	R/W	TSNTag 中的“send addr”	0
.....				

send_table_1023	16	R/W	保留	0
	15	R/W	表项有效位, 0 代表无效, 1 代表有效	0
	14:5	R/W	ST 流在一个应用周期内的注入时间槽	0
	4:0	R/W	TSNTag 中的“send addr”	0
0x100400-0x1fffff			保留	

## ● TSS 模块

地址范围为 Addr 0x200000-0x2fffff。

表 C-7 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
submit_table_N 0x200000-0x2003ff	ST 报文提交时刻表每项内容, N=0、1、...、1023 submit_table_0 表示第 0 个提交表			
0x200400-0x2fffff	保留			

表 C-8 寄存器的具体含义

name	bit	R/W	description	default
submit_table_0	31:16	R/W	保留	0
	15	R/W	表项有效位, 0 代表无效, 1 代表有效	0
	14:5	R/W	ST 流的提交时间槽	0
	4:0	R/W	TSNTag 中的“send addr”	0
.....				
submit_table_1023	31:16	R/W	保留	0
	15	R/W	表项有效位, 0 代表无效, 1 代表有效	0
	14:5	R/W	当前 Slot	0
	4:0	R/W	TSNTag 中的“send addr”	0

## ● QGC 模块

地址范围为 Addr 0x300000-0xafffff, 其中 0x300000-0x3fffff 表示第一个端口的门控表, 以此类推, 共有 8 个端口门控。

表 C-9 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
port0_gate_table_N 0x300000-0x3003ff	0 号端口的门控表, N=0、1、...、1023, 输出门控 port0_gate_table_0 表示 0 号端口的第一个时刻的门控状态			
port1_gate_table_N 0x400000-0x4003ff	1 号端口的门控表, N=0、1、...、1023			
port2_gate_table_N 0x500000-0x5003ff	2 号端口的门控表, N=0、1、...、1023			
port3_gate_table_N 0x600000-0x6003ff	3 号端口的门控表, N=0、1、...、1023			
port4_gate_table_N 0x700000-0x7003ff	4 号端口的门控表, N=0、1、...、1023			
port5_gate_table_N 0x800000-0x8003ff	5 号端口的门控表, N=0、1、...、1023			
port6_gate_table_N 0x900000-0x9003ff	6 号端口的门控表, N=0、1、...、1023			
port7_gate_table_N 0xa00000-0xa003ff	7 号端口的门控表, N=0、1、...、1023			

表 C-10 寄存器的具体含义

name	bit	R/W	description	default
port0_gate_table_0	31:8	R/W	保留	0
	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态, 0 代表该队列的门控关闭, 1 代表开启	0
.....				
port7_gate_table_1023	31:8	R/W	保留	0
	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态, 0 代表该队列的门控关闭, 1 代表开启	0

### ● FLT 模块

地址范围为 Addr 0xc00000-0xcfffff。

表 C-11 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
0xc00000-0xc03fff	forward_table_N, 表示转发表, N=0,1,2, ...16383, forward_table_0 表示第 0 个转发表			
0xc04000-0xcfffff	保留			

表 C-12 寄存器的具体含义

name	bit	R/W	description	default
forward_table_0	31:16	R/W	保留	0
	8:0	R/W	转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值 0 代表不向该端口转发，1 代表向该端口转发	0
.....				
forward_table_16383	31:16	R/W	保留	0
	8:0	R/W	转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值为 0 代表不向该端口转发，1 代表向该端口转发	0
0xc04000-0xcffff			保留	

## 附录 D：NMAC 报文格式

### ● 配置报文格式

配置报文格式：在报文中用 **count** 字段（8bit）表示报文中包含的配置条目数，报文最小为 64 字节，最后不够 64 字节的报文需要补零。NMAC 命令在以太网报文中的封装如图 D-1 所示。

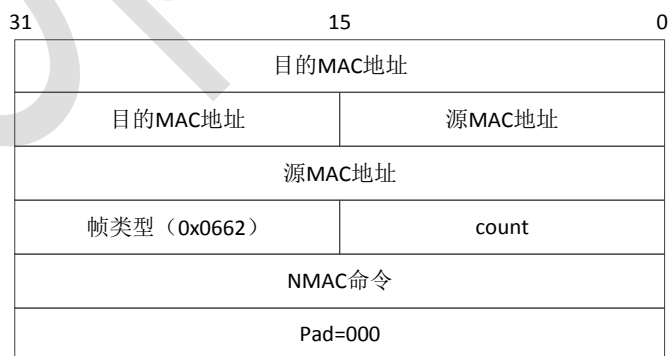


图 D-1 NMAC 配置报文格式

NMAC 命令的格式如图 D-2 所示。

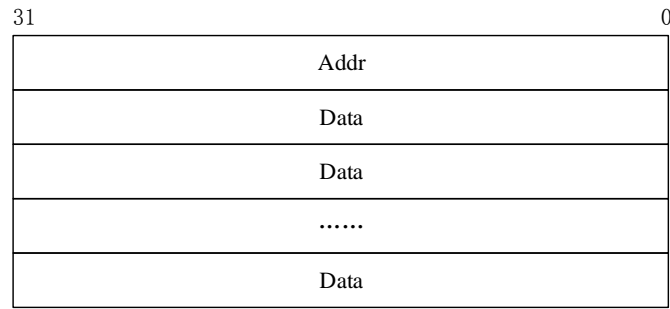


图 D-2 NMAC 命令格式

当配置的寄存器数量为 1，NMAC 命令就包括 32bit 的 ADDR 和 32bit 的 DATA；当配置的寄存器数量为 N（N>1），NMAC 命令就包括 32bit ADDR 和 N\*32bit 的 DATA，第一个 DATA 以 ADDR 作为 RAM 写地址，第二个以及后续 DATA 以 ADDR 循环加 1 作为 RAM 写地址。

● 上报报文格式

上报报文格式：其中报文类型为 NMAC 报文（0x1662）。

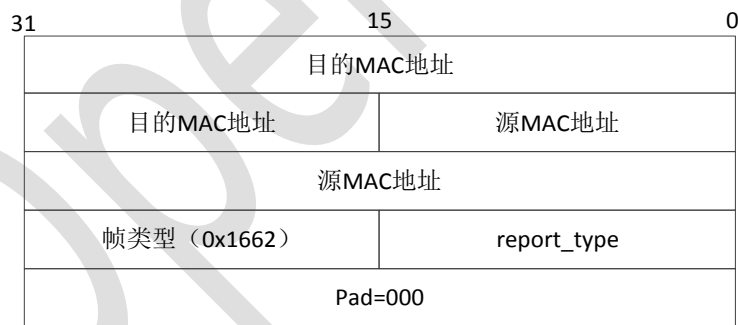


图 D-3 NMAC 上报报文格式

表 D-1 上报类型格式

上报类型（16bit）		含义
高 6bit	低 10bit	
000000 单个寄存器	0	配置的单个寄存器，包含配置完成寄存器、端口状态寄存器、时间槽大小寄存器、时间偏移寄存器、上报周期寄存器、上报类型寄存器、应用周期寄存器



上报类型（16bit）		含义
高 6bit	低 10bit	
000001 转发表	0	第 0-63 条转发表
	1	第 64-127 条转发表
	2-255	第 128-16383 条转发表
000010 注入时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000011 提交时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000100-001 011 P0-P7 输出 门控表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
001100 xx_state	0	暂未开发

单个寄存器，report\_type 高 6bit 为 0000，低 10bit 为 0。

转发表上报报文格式，report\_type 高 6bit 为 000001，低 10bit 为上报的第几块，转发表一共有 16K 条，每条转发表占用 2 字节（其中低 9bit 为有效数据），因此每个报文可以携带 64 条，总共需要 256 个报文。

注入时刻表上报报文格式，report\_type 高 6bit 为 000010，低 10bit 为上报的第几块，注入时刻表一共有 1024 条，每条转发表占用 2 字节（其中低 9bit 为有效数据），因此每个报文可以携带 64 条，总共需要 16 个报文。

提交时刻表上报报文格式，report\_type 高 6bit 为 000011，低 10bit 为上报的第几块，注入时刻表一共有 1024 条，每条转发表占用 2 字节（其中低 9bit 为有效数据），因此每个报文可以携带 64 条，总共需

要 16 个报文。

门控表按照端口划分，每个门控占用一块 RAM，共有 8 个端口。

每个端口两块 RAM，总共需要 16 块 RAM。

## 附录 E：command/command\_ack 命令格式

表 E-1 command/command\_ack 命令格式

位置	位宽	名称	说明
[203:180]	8	node_id	该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。
[179:172]	8	dest_module_id	该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID
[171:168]	4	type	4'b0001:寄存器或表项的写命令； 4'b0010:寄存器或表项的读命令； 4'b0110:寄存器或表项的读响应。
[167:152]	16	addr	寄存器或表项的读/写地址
[151:0]	152	data	寄存器或表项的读/写数据；其中五元组映射表的表项位宽最大，为 152bit